UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/997,163	11/29/2001	Giuseppe Desoli	10011614-1	3624

22879 7590 08/26/2004

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER	
PHAM, CHRYSTINE	
ART UNIT	**PAPER NUMBER**
2122	

DATE MAILED: 08/26/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/997,163	DESOLI ET AL.	
	Examiner	Art Unit	
	Chrystine Pham	2122	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *29 November 2001*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-36* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-36* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *29 November 2001* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Specification

1. The lengthy specification has not been checked to the extent necessary to determine the presence of all possible minor errors. Applicant's cooperation is requested in correcting any errors of which applicant may become aware in the specification.

Double Patenting

2. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

3. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. See *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970);and, *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent is shown to be commonly owned with this application. See 37 CFR 1.130(b).

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

This is a <u>provisional</u> obviousness-type double patenting rejection.

4. Claims 1-3, 5-7, 10, 12-15, 17-21, 23, and 30 are provisionally rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1, 8-10,

12-16, 18, 20-21, and 28 of copending Application No. 09999451 (hereinafter ***CA1*** whose references will be cited in bold text) in view of Buzbee et al. (US 5933622), hereinafter, *Buzbee et al.*

As per claim 1, ***CA1*** discloses a method (see **claim 1**) and system (see **claim 10**) for

- fetching program code (see **claim 1 line 2**);
- translating program code (see **claim 1 line 4**);
- emitting translated program code into at least one code cache (see **claim 5 line 2**, and **claim 12 line 2**); and
- executing translated code within the at least one code cache in lieu of associated program code when a semantic function of the associated program code is requested (see **claim 12 line 1-2**).

CA1 does not expressly disclose executing a program written for an original computer system on a different host computer system. However, *Buzbee et al.* teach a method (e.g., col.1:5-10) and system (e.g., col.2:65) for executing a program written for an original computer system (e.g., see Abstract *first computer*) on a different host computer system (e.g., see Abstract *second computer*), comprising the steps of:

- fetching and dynamically interpreting and translating program code into code fragments (e.g., *blocks of code* col.3:16-17) with an interpreter/emulator (e.g., col.1:28-31 & 33-35, col.2:65-67, col.3:) ;
- emitting translated program code (i.e., code fragments) into at least one code cache (e.g., col.316-19) within a virtual machine (e.g., col.1:39-45); and
- executing translated code within the at least one code cache in lieu of associated program code when a semantic function of the associated program code is requested (e.g., col.316-19).

Buzbee et al. further teach the steps of emulating actions which would have been performed by the original computer system during execution (e.g., see Abstract, col.1:39-48,

col.2:25-28, col.3:29-34), translating program code/instructions with an optimizing compiler (e.g., see Abstract, col.4:42-62), and tracking cached code fragments using associated metadata (e.g., see *key addresses* col.3:18-19). It would have been obvious to one of ordinary skill in the pertinent art at the time the invention was made to combine the teaching of *Buzbee et al.* with that of *CA1* to obtain a method and system for executing a program written for an original computer system on a different host computer system which would produce the expected result with reasonable success. And the motivation for doing so would have been to minimize the cost of translating code from an original computer to host computer having a different instruction set and to eliminate the need to obtain the source code from the original computer (for recompilation) via an emulator which is capable of providing a virtual machine which emulates the original computer in handling asynchronous events and dynamically translating original computer's code at runtime.

As per claim 5, *CA1* as modified by *Buzbee et al.* discloses a method as applied to claim 1, wherein the step of emitting translated program code into at least one code cache comprises emitting translated program code into the at least one code cache via an application programming interface (see **claim 28 line 4-5**).

As per claim 6, *CA1* as modified by *Buzbee et al.* discloses the method as applied to claim 1 above, further comprising the step of interpreting and executing program code that has not be emitted into the at least one code cache (see claims 8-9 in their entirety).

As per claims 2-3, 7, 10, and 14, see claim 1 above.

As per claim 15, *CA1* as modified by *Buzbee et al.* discloses an emulation program (see *decoding program* **claim 16 line 1**) configured to emulate an original computer system for which a program was written, the emulation program stored on a computer-readable medium (see *computer-readable medium* **claim 16 line 1**) and comprising:

- logic configured to translate program code (see **claim 16 line 6**);

- logic configured to emit code fragment translations of program code into at least one code cache (see **claim 18 line 2**, claim **20 line 1-2**); and
- logic configured to execute the code fragments within the at least one code cache in lieu of associated program code when a semantic function of the associated program code is requested (see **claim 18 line 1-2**).

As per claim 20, *CA1* as modified by *Buzbee et al.* discloses a system for executing program code that was written for an original computer system on a different host computer system, comprising:

- an emulator (see claim 1 above);
- a translator (see claim 1 above);
- a virtual machine that comprises a dynamic execution layer interface including a core having at least one code cache in which code fragments can be cached and executed (see claim 1 above); and
- an application programming interface (see *application programming interface* **claim 28 line 4**) that links the translator to the virtual machine (see **claim 28 line 4-5**).

As per claims 12-14, 17-19, 21, 23 see claims 1, 5-6 above.

As per claim 30, *CA1* teaches an application programming interface (see **claim 28 line 4**) configured to link a translator to a dynamic execution layer interface (see *dynamic execution layer interface* **claim 28 line 1-5**) in an computer system emulating system, comprising:

- a set of functions available to the translator (see claim 1 above) including:
  - an emit fragment function with which the translator can emit code fragments into code caches of the dynamic execution layer interface (see **claim 28 line 3**) , and

- an execute function with which the translator can request execution of code fragments contained within the at least one code cache (see **claim 28 line 4-5**).

Claim Rejections - 35 USC § 102

5. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

6. Claims 1-3, 10, and 14 are rejected under 35 U.S.C. 102(b) as being anticipated by Buzbee et al. (US 5933622), hereinafter, *Buzbee et al.*

As per claim 1, *Buzbee et al.* teach a method (e.g., col.1:5-10) and system (e.g., col.2:65) for executing a program written for an original computer system (e.g., see Abstract *first computer*) on a different host computer system (e.g., see Abstract *second computer*), comprising the steps of:

- fetching and dynamically interpreting and translating program code into code fragments (e.g., *blocks of code* col.3:16-17) with an interpreter/emulator (e.g., col.1:28-31 & 33-35, col.2:65-67, col.3:) ;
- emitting translated program code (i.e., code fragments) into at least one code cache (e.g., col.316-19) within a virtual machine (e.g., col.1:39-45); and
- executing translated code (by translator) within the at least one code cache in lieu of associated program code when a semantic function of the associated program code is requested (e.g., col.3:14-19).

Buzbee et al. further teach the steps of emulating actions which would have been performed by the original computer system during execution (e.g., see Abstract, col.1:39-48, col.2:25-28, col.3:29-34), translating and linking program code/instructions with an optimizing

compiler (e.g., see Abstract, col.4:42-62), and tracking cached code fragments using associated metadata (e.g., see *key addresses* col.3:18-19).

As per claims 2-3, 10, and 14, they recite limitations which have been addressed in claim 1, therefore, are rejected for the same reasons as cited in claim 1.

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

8. Claims 4, 8-9, 11 are rejected under 35 U.S.C. 103(a) as being unpatentable over *Buzbee et al.* as applied to claims 1 and 10 above, further in view of Lethin et al. (US 6463582), hereinafter, *Lethin et al.*.

As per claim 4, *Buzbee et al.* teach the method as applied to claim 1, wherein the step of translating the program code comprises translating program instructions with an optimizing compiler. *Buzbee et al.* do not expressly disclose the optimizing compiler as being just-in-time (JIT) compiler. However, *Lethin et al.* discloses an optimizing compiler as being the JIT compiler/translator (e.g., see Abstract, FIG.24,25 & associated text). *Lethin et al.* further disclose the JIT compiler growing a code segment by linking program instructions together prior to emitting translated program code (e.g., col.2:10-15, see FIG.2,8,9,10 & associated text) and an execute function within an application programming interface with which the translator can request execution of code fragments contained within the at least one code cache (e.g., FIG.3,11 & associated text, col.16:55-col.17:14). It would have been obvious to one of ordinary skill in the

pertinent art at the time the invention was made to modify the teaching of *Buzbee et al.* to include the JIT compiler as disclosed by *Lethin et al.*, which would produce the expected result with reasonable success. And the motivation for doing so would have been that traditional compilers would need to obtain and compile the source code to produce the corresponding executable code. The downside of traditional compilers is that the source code might not be obtainable to a new computer system with a different instruction set, and traditional compilers produce executable code which is platform-dependent. JIT compilers would have been preferred instead as being capable of determining program methods/instructions which are called most often, interpreting and translating the associated byte code at runtime to produce platform-independent executable code for the instructions.

As per claims 8-9, and 11, they recite limitations which have been addressed in claim 4, therefore, are rejected for the same reasons as cited in claim 4.

9. Claims 5 and 12 are rejected under 35 U.S.C. 103(a) as being unpatentable over *Buzbee et al.* as applied to claims 1 and 10 above, further in view of Challenger et al. (US 6256712), hereinafter, *Challenger et al.*

As per claim 5, *Buzbee et al.* teach method of claim 1. *Buzbee et al.* do not expressly disclose emitting translated program code into the at least one code cache via an application programming interface. However, *Challenger et al.* disclose emitting data into the at least one code cache (e.g., see *cache 2* FIG.1A & associated text) via an application programming interface (API) (e.g., see *cache manager 1 & API's* FIG.1A & associated text) which is interacted by a translator/compiler (e.g., col.27:50-55, FIG.5 & associated text). *Challenger et al.* further disclose the API functions including:

- the emit fragment function (e.g., see *410* FIG.4 & associated text) can be used to perform at least one of tracking a cached object and associating metadata (e.g., see *object_id* FIG.4 & associated text) with a cached object.
- a lookup fragment function with which cached objects can be retrieved (e.g., see *415* FIG.4 & associated text).
- an invalidate fragment function with which individual cached objects can be invalidated (e.g., see *450* FIG.4 & associated text).
- a cache flush function in which caches can be flushed (e.g., col.1:30-35, col.5:41-42, see *410* FIG.4 & associated text).
- an install callback function with which the translator can be notified as to events that occur within the cache management (e.g., see *return status 1240* FIG.6 & associated text, see *1680* FIG.7 & associated text, see *1150* FIG.8 & associated text, see FIG.9-11, 17 & associated text).
- a enumerate fragment function with which cached code fragments can be numerated using associated metadata (e.g., see *470* FIG.4 & associated text).

It would have been obvious to one of ordinary skill in the pertinent art at the time the invention was made to combine the teaching of *Buzbee et al.* with that of *Challenger et al.* to obtain an API which mediates between a translator and the at least one code cache since APIs, in general, are known to provide a level of abstraction between the application and the its functions/utilities/services to ensure portability/reusability of the code. Furthermore, APIs are designed to encapsulate complex implementation/coding of the API's functions, thus providing a streamlined interface to other applications/programs/software using the application. Other applications (e.g., translator) can then utilize the application's services/functions (e.g., emit fragment, invalidate fragment) via invoking the methods exposed in the interface (e.g., cache manager API), without undue modification of the other applications' codes.

As per claim 12, it recites limitations which have been addressed in claim 5, therefore, is rejected for the same reasons as cited in claim 5.

10. Claims 6-7, 13 are rejected under 35 U.S.C. 103(a) as being unpatentable over *Buzbee et al.* as applied to claims 1, and 10 above, further in view of Morley (US 5751982), hereinafter, *Morley*.

As per claim 6, *Buzbee et al.* teach the method of claim 1 further comprising the step of interpreting program code that has not been emitted into the at least one code cache (see claim 1). *Buzbee et al.* do not expressly disclose the step of executing program code that has not been emitted into the at least one code cache. However, *Morley* discloses an emulation method and program (e.g., col.1:5-12, col.3:32-33) for emulating an original computer system for which a program was written (e.g., see Abstract, col.3:33-35), the emulation program stored on computer-readable medium (e.g., col.3:37-44), the method/program comprising the step of executing program code that has not been emitted into the at least one code cache (e.g., col.1:44-49 & 60-65, col.3:65-col.4:6). It would have been obvious to one of ordinary skill in the pertinent art at the time the invention was made to combine that teaching of *Morley* with that of *Buzbee et al.* to include the step of executing program code that has not been emitted into the at least one code cache which would produce the expected result with reasonable success. And the motivation for doing so would have been that the capability to execute program code which are not emitted to code caches allows programmers/designers of the emulation system the versatility of storing program code in various different types of memory (e.g., main memory, cache memory, and/or permanent storage) within the emulation system depending on their choices and/or the particular architecture of said emulation system.

As per claims 7, and 13, they recite limitations which have been addressed in claim 6, therefore, are rejected for the same reasons as cited in claim 6.

11. Claim 15 recites limitations which have been addressed in claim 6, therefore, is rejected for the same reasons as cited in claim 6.

12. Claim 16-19 recite limitations which have been addressed in claims 4-6, therefore, are rejected for the same reasons as cited in claims 4-6.

13. Claims 20-29 are rejected under 35 U.S.C. 103(a) as being unpatentable over *Buzbee et al.* further in view of *Lethin et al.* and *Challenger et al.*.

As per claims 20-29, they recite limitations which have been addressed in claims 4-5, therefore, are rejected for the same reasons as cited in claims 4-5.

14. Claim 30 is rejected under 35 U.S.C. 103(a) as being unpatentable over *Buzbee et al.* further in view of *Challenger et al.*.

As per claim 30, it recites limitations which have been addressed in claim 5, therefore, is rejected for the same reasons as cited in claim 5.

15. Claims 31-36 recite limitations which have been addressed in claims 4-5, therefore, are rejected for the same reasons as cited in claims 4-5.

Conclusion

16. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure:

- Emulation devices, systems and methods utilizing state Machines, Swoboda et al. (US 5329471)
- Uncoupling a central processing unit from its associated hardware for interaction with data handling apparatus alien to the operating system controlling said unit and hardware, Baker et al. (US 5388215)

- Method of using a target processor to execute programs of a source architecture that uses multiple address spaces, Scalzi et al. (US 5560013)
- Operating system for embedded computers, Keeley (US 6138271)
- Method and apparatus for caching the results of function applications with dynamic, fine-grained dependencies, Heydon, Clark Allan et al. (US 6145056)

17. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Chrystine Pham whose telephone number is 703.605.1219. The examiner can normally be reached on Mon-Fri, 8:30am-5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q Dam can be reached on 703.305.4552. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Chrystine Pham
Examiner
GAU 2122

TUAN DAM
SUPERVISORY PATENT EXAMINER